下载APP

24丨技术观：程序员在技术的成长之路上，有哪些陷阱？

2020-07-10 臧萌

职场求生攻略

进入课程 >

讲述：臧萌

时长 15:04　大小 13.80M

你好，我是臧萌。今天，我们对上一节课做一下延伸，基于上一节课中抛出来的技术观，今天我们来看一下在我们实际工作中，关于技术的事儿都有哪些陷阱。

为了给你增加一些代入感，我们首先来看一个厨师的例子。假设你是一个餐厅的老板，要招一位厨师。

厨师小白来面试了，土豆丝切得要么像吸管一样粗，要么像绣花针一样细。炒出来的土豆丝一盘偏咸，一盘偏淡，仔细一看土豆皮还没削干净。

厨师小稳来面试了，只见他麻利地抓起土豆，开始削皮切丝，青椒去瓤切条，大火爆炒。10 分钟后，热气腾腾香气扑鼻的土豆丝就出锅装盘了，一尝，嗯，标准的饭馆儿味。小稳憨厚一笑，道：“如果有干红辣椒，用油一爆，那就更棒了。”

厨师小洁来面试了，对着后厨一堆菜打量了半天，眉头拧成个疙瘩，嘴上啧啧叹道：“这个土豆不行呀，大的大小的小；这辣椒也不行，得是刚刚摘下来的；油也不合适，得是古法压榨的菜籽油。”吐槽了半天，像是受了天大的委屈一样，磨磨唧唧半个小时才“勉为其难”地炒了一盘土豆丝，上口一尝，味道只能算一般。小洁皱着眉头地说道：“这些材料不大行。”

厨师小高来面试了，把土豆削一削切一切，一斤土豆就用半斤。土豆丝呢，切得倍儿棒，每根都是 1mm 粗，6cm 长，误差不超过 0.1mm，堪称艺术品。土豆丝炒好之后，小高把土豆丝一根根地码到盘子里，一“根”不乱。3 小时后，凉透的土豆丝上桌了。

厨师老遗来面试了，进后厨第一句话就是问你：“咱这儿有土灶吗？烧柴火、拉风箱的那种，我用得惯。”你遗憾地表示，这个真没有，以后也不大可能会有。老遗叹了口气扭头走了，边走边念叨：“燃气灶烧出来的菜不香。”

现在你是老板，你会选择雇用哪个厨师呢？厨师嘛，自然要有刀工、配菜、烹饪、装盘等技术。但是最终的目的，就是要利用一切合理的资源，做出色香味均衡的菜肴给食客。

所以小稳自然是没得说。

剩下四位的毛病，相信你也已经心里有数。其实他们身上体现的问题，就是我总结的技术人常犯的四种错误：

1. 技术不扎实，不思进取；
2. 技术洁癖；
3. 一味追求新潮流，新技术；
4. 固步自封。

下面我就谈谈这四种错误该如何纠正。当然，我要强调一点，不管是哪种问题，都得充分发挥自己的主观能动性。需要你由内而外地纠正，自己认识到错误，发自内心地想改正。

技术不扎实，不思进取

软件行业的“小白厨师”，技术能力已经差到无法产生有效生产力的程度。

事实上，“小白”是一个相对概念，不是绝对概念。也就是说，在人生的不同阶段，我们都有可能是一个“小白”。

大学刚刚毕业的天之骄子，到公司变成一个普通员工，是小白；从之前的业务骨干，转到新的工作方向，也是小白。小白并不可怕，怕的是一直“白”下去。如果没有危机意识，惯性地以为自己可以凭借之前的地位和高度，就能够完全掌握新的工作内容，这是很危险的。

当我们换一个工作或者方向的时候，都是从小白开始，这时候的我们生产力很低，甚至为负。我们一定要意识到并且承认这一点，积极主动学习、补齐短板，至少做到合格。那么成为小稳，指日可待。

技术洁癖，满眼问题

软件工程师中的“小洁”，满眼都是问题，看到的都是各种先天不足。没错，每个公司，每个项目都有现实的问题。也许这个项目不能用 Oracle 只能使用 MySQL；也许这个项目不得不临时调走几个人，导致团队人变少了；也许这个项目不能使用 Spark 只能用老掉牙的 Hive；也许这个项目没有预算买 Zing JDK……

如果只能依赖先天优势，看到不足就无从下手，那我们的工作还有什么意义？我们的价值又从何处体现？

身为软件工程师，我们拿这份工资，就是要解决这些问题的，而不是去一味地抱怨。软件工程师就是要在现有的条件下，结合自己的技术，解决问题，把事情办成，让项目落地。当然，并不是所有问题都能靠软件工程师解决，但是能解决到什么程度，就很考验一个软件工程师的功力了。

项目在落地的过程中，外部条件不如意者，十之八九。软件工程师的职责就是要找出关键点，给出合理的期待。

如果没有 Oracle，就找方案，让数据在 MySQL 上也能有足够的安全性和可靠性；如果临时有人被抽调走，就把项目排出个轻重缓急，不重要的事情先放一放；如果不能用 Spark，就看看是否可以把数据刷新的速度从小时级变为天级；如果没有 Zing JDK 的加持，就试试把 P99 增加到 500ms，如果这个数值不理想，可以后期调优。把这些数据和相关人等沟通好、协调好，并得出一个可行的方案，才是软件工程师的价值所在。

小稳厨师就没追求吗？他也想炒出更完美的炒土豆丝。但是他知道，有没有干辣椒，并不是炒土豆丝的必要条件。你作为老板，面对小稳圆满完成工作之后提出的请求，和小洁什么都没做就提出的一大堆请求，你会考虑满足哪个的要求呢？答案不言自明。小稳已经证明了自己，老板当然会选择给他更多信任。

张口闭口就是“这也不行，那也不行”的人，其实根子上，就是自己解决问题的能力不行。

避免使用不必要的技术手段

这个观点的意思其实很明确，就是当你在工作中使用一项技术，要用保守的态度，要以投入产出比做衡量，而不是根据自己对技术的偏好和冲动乱来。小高和老遗就是正反两个极端。

我们来看看这两种态度。

学了新技术，内心很骚动，不用不爽

首先是小高厨师，看得出来小高厨师是有点骄傲的，程序员也都是有点骄傲的。感觉我们技术人就像魔法师，看不起不会技术的“麻瓜”。而学会新技术的人，又按耐不住显摆一下的冲动。

“学了本领，不显摆显摆别人怎么知道我会呢？”于是在这种心境下，就很容易小题大做，杀鸡拿起宰牛刀。有些技术不错的年轻人，就是因为做事情的态度有问题，才导致自己发展受限。

比如说，Scala 这个语法糖好甜，我要把项目都转成 Scala 的。Guava 新版里这个 API 看着不错，我用用看。

结果呢？语法糖是糖衣“坑”弹，各种想当然的接口和 Java 并不兼容；Guava 是 Google 的，Google 是业内典型的视兼容性为无物的公司，新版 Guava 和老版 API 不兼容，只能回退。当然还有更骚的，比如使用库中 internal 的 API；使用实验性质的 JVM 功能；将系统架构在一个连文档都没有的新功能上等等等等。

各种骚操作，不一而足。炫技一时爽，全组陪你加班到天亮。

其实很多时候，觉得自己一身本领无处施展，主要还是因为自己学艺不精，没学到位，不知道怎么正确地施展。下面我们就看看，怎么拯救炫技的厨师小高。

小高无疑是对技术有追求的，刀工不错。但是他浪费了一半的土豆，而且一道本可以 10 分钟就搞定的菜，生生用了 3 个小时。小高的问题是，本末倒置，而且功夫没到家。

用户要的是一道炒土豆丝。速度快，味道好，这是红线，不可逾越。小高要做的，就是精进自己的技术，在保证十分钟可以搞定的前提下，再保证土豆丝一丝不乱的造型，不浪费那么多土豆。这时候，小高的这道炒土豆丝将会是饭店的一大特色。否则，就是一道摆设了。

回到我们在上文中举的例子，面对这些情况，又该怎么办呢？

Scala 可以用，前提是吃透，能提高大家效率，不增加集成工作量。Guava 可以升级，但是要先好好阅读文档，检查兼容性问题和整个项目的依赖关系，确保升级没有问题。而内部 API 这些，真的不要主动用，只有在救火的时候，才可以作为临时的解决方案。

再举一个例子。我在 12 年的时候买了一个谷歌 Neuxs 3，那时候它就支持前置摄像头面部识别解锁。但是体验是怎么样的呢？嗯，10 次里面大概有 3 次是失败的，更不要提光线和角度不完美的情况。直到 5 年后，也就是 2017 年，苹果才推出了成熟的面部解锁方案。

所以说，一个技术的出现和一个技术可以用在生产上，完全是两个概念。一个技术在成熟之前，在确定能给用户带来价值之前，不要盲目使用。不要沉浸在自己的技术世界里不可自拔。谷歌不差钱，可以搞酷炫的东西，培养自己技术急先锋的人设，但是，普通公司并不愿意付出这个成本。

其实很多时候，老板宁愿雇佣小白，也不愿意雇佣小高。只要愿意干，技术不行可以练，不能炒菜可以先练习切菜、配菜，但如果做事情的态度出了问题，很难由外而内地改。但这一点也只能由小高自己去领悟（想让老板容忍一盘土豆丝炒了三个小时，除非厨师是老板的小舅子）。

在用宰牛刀杀鸡之前，不妨先问问自己，宰牛刀法练好了吗？

新技术要好好学，吃透了，才能用得对，用得好。用之前问问自己，使用这个技术的代价是什么？价值是什么？事情值得做吗？我是为了解决实际的需求，还是为了满足自己那颗躁动的心？

固步自封的遗老遗少

和小高相反，老遗厨师代表的另一个极端——固步自封。用自己 N 年前的观点，衡量现在的事物，拒绝接受事物的发展，对技术存有刻板偏见，倚老卖老。

举个例子，我有个朋友跟我吐槽，说他的经理眼里只有 C++，看不起 Java，张口闭口就是 Java 慢，Java 垃圾，他当初写 C++ 怎么怎么地。听说 Impala 是用 C++ 写的，就找人研究，要替换公司里的 Hive，又听说百度用 C++ 重写了一遍 Hadoop，就又来了精神。可能他觉得自己有金手指吧，一个手指就能搞定数据迁移，客户端升级和测试。

我听到后，笑得都合不拢嘴了。必须承认 C++ 是一门优秀的技术，我认识很多 Java 的大牛，本身就是 C++ 的大牛。技术本身没有好坏，只有合适不合适。既成事实是，当今整个基于 JVM 的大数据生态既完整，又成熟稳定。而且公司在有一套 Hadoop 平台的前提下，只有一个语言的遗老才可能因为语言原因，想要去替换平台。

据说等他离职了，也没完成他的夙愿。

技术是会发展的，以前有的缺点，不代表未来一直都有。对技术不要有无谓的偏见，技术就是工具，就是用来解决问题的。一项技术能够在行业中流行，必定是有原因的。若是固步自封，早晚会被淘汰。现在还在念叨“燃气灶烧出来的菜不香”的厨师，已经不适合在饭店做厨师了，可能农家乐里还有他们的一丝生存空间。

总结

中国有句话，叫学好数理化，走遍天下都不怕。其实我觉得应该说成“用好数理化，走遍天下都不怕”。

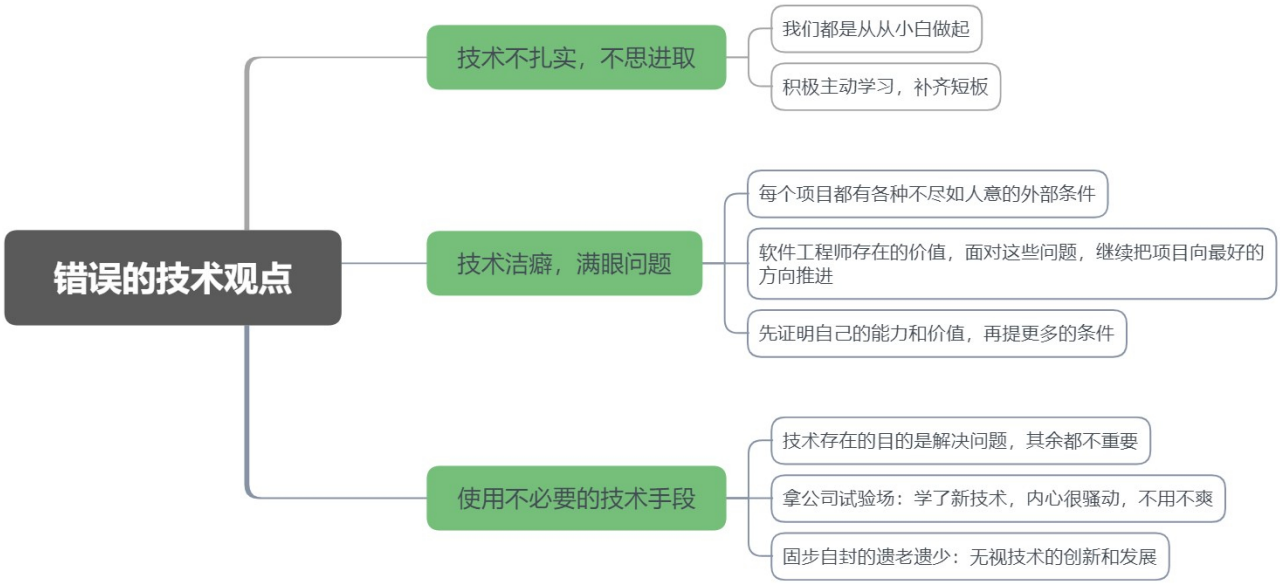
软件工程师靠技术吃饭，但是世界并不是绕着技术转的。我们要对技术要有追求，但是要分清什么是自己的追求，什么是公司的追求。除了极少数脱离了低级趣味的公司之外，绝大多数公司雇佣软件工程师的目的，是为了创造价值。

技术是手段而非目的，炫技很没必要，固步自封必被淘汰。技术的目的是解决问题，而非带来更多的问题。

当然很多公司会鼓励大家学习新技术，还会举办类似“编程马拉松”的活动。这是因为技术公司基本都看到了，如果新技术用得好，很多时候可以带来生产力的提升。这么做也是变相地做人才和技术储备。公司鼓励你抬头看天，但是并不鼓励你拿着公司的钱，坐热气球去看天。

态度只能由内而外地改变。如果不是自己认识到问题，是不可能积极主动地改正的。不要让自己对技术错误的态度，成为职业生涯的绊脚石和阻力。

当然，你可能会有疑问：说得这么冠冕堂皇，自己真的能做到吗？确实人都有不自觉的时候，但是做得不对，和不认为自己做得不对，是两回事儿。我们要做的就是时常审视自己。吾日三省吾身。不怕自己犯错，只怕犯错不改。



极客时间

思考题

最后问你几个小问题。在自己成长的路上，你有没有经历哪个阶段，是文中的小白、小稳、小洁、小高或者老遗呢？你现在对技术的态度，和这篇文章有没有冲突呢？你还见过哪些技术路上的陷阱呢？欢迎你在评论区讨论。

好，今天的课程就结束了，希望可以帮助到你，也希望你在下方的留言区和我参与讨论，并把文章分享给你的朋友或者同事，一起交流一下。

提建议

更多课程推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 23 | 技术观：做程序员，技术观为何如此重要？

下一篇 25 | 系统架构：如何从写代码的程序员，成长为软件系统架构师？

精选留言 (3)

写留言

sugar

2020-07-10

每个大厂都会有少数部门做着技术研究性的工作，这些team是为数不多的可以不碰业务少写脏代码的，极端性的例子应该是当年赫赫有名的bell lab。如今国内像BAT这样的成熟公司也会设有类似的部门和团队，请教老师对技术研究性的工作怎么看？什么样的人适合进这样的team（在本身技术能力过硬，有的选的情况下）去走纯技术路线？这条路走下去天花板如何，国内环境走这样的路线前景如何（毕竟国内年轻化 青春饭问题叫人硅谷而言...

展开

作者回复: 我觉得真正适合走这种路线的还是比较少的。完全和业务没关系，很难证实自己创造的价值。另一方面，对于职业来说，选择大于努力。

举个例子，Yahoo的技术曾经也是很厉害的，包括Hadoop，很多前端技术，都是Yahoo主导的。

如果想走研究方向，还是读博走学术道路比较靠谱。

5

Sdylan
2020-07-10

学再多的技术不用，瞎忙活。只有帮用户解决问题，才是选对了技术。技术本身没有好坏之分，有人用在不好的场景才会说技术不好。就如你切菜，非等选锤子来整，是锤子不行还是人不行，不言而喻。

作者回复: ✅ ✅

2

牛牛
2020-07-10

在职场政治上、我是小白、拿对待朋友的那一套来对待工作、很多时候总觉得哪里不对、又想不明白、老师无疑给我补上了这一课、手动送花花 ~

展开

作者回复:
很高兴能对你有帮助 ~

1